

ORACLE®

Oracle Digital Assistant

The Complete Training

Custom Components Overview

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Topic agenda

- 1 Overview
- 2 Custom component architecture
- 3 Deployment options
- 4 Design considerations

Topic agenda

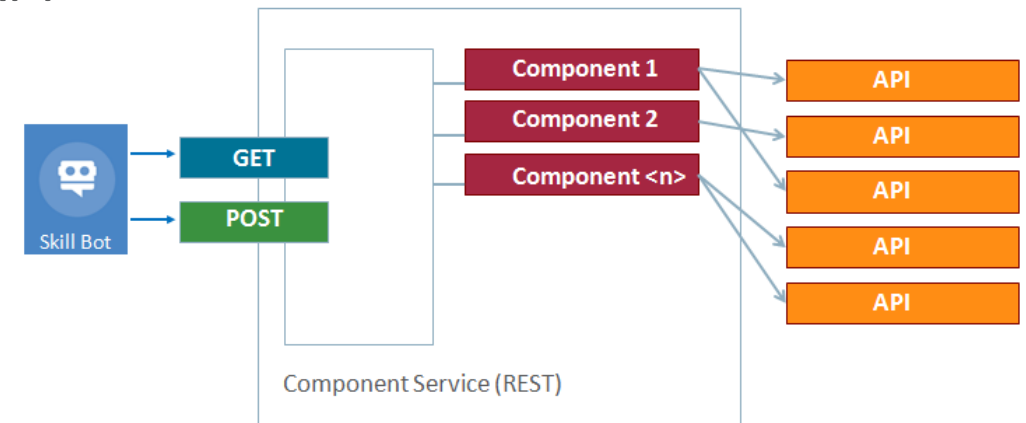
- 1 Overview
- 2 Custom component architecture
- 3 Deployment options
- 4 Design considerations

About custom components

- Like system components, just custom
- Use cases
 - Backend system calls
 - Integrate custom code logic into a bot conversation
- Developed using Node.js
- Hosted in custom component services
 - One or many components per service
 - REST Service built with Node Express

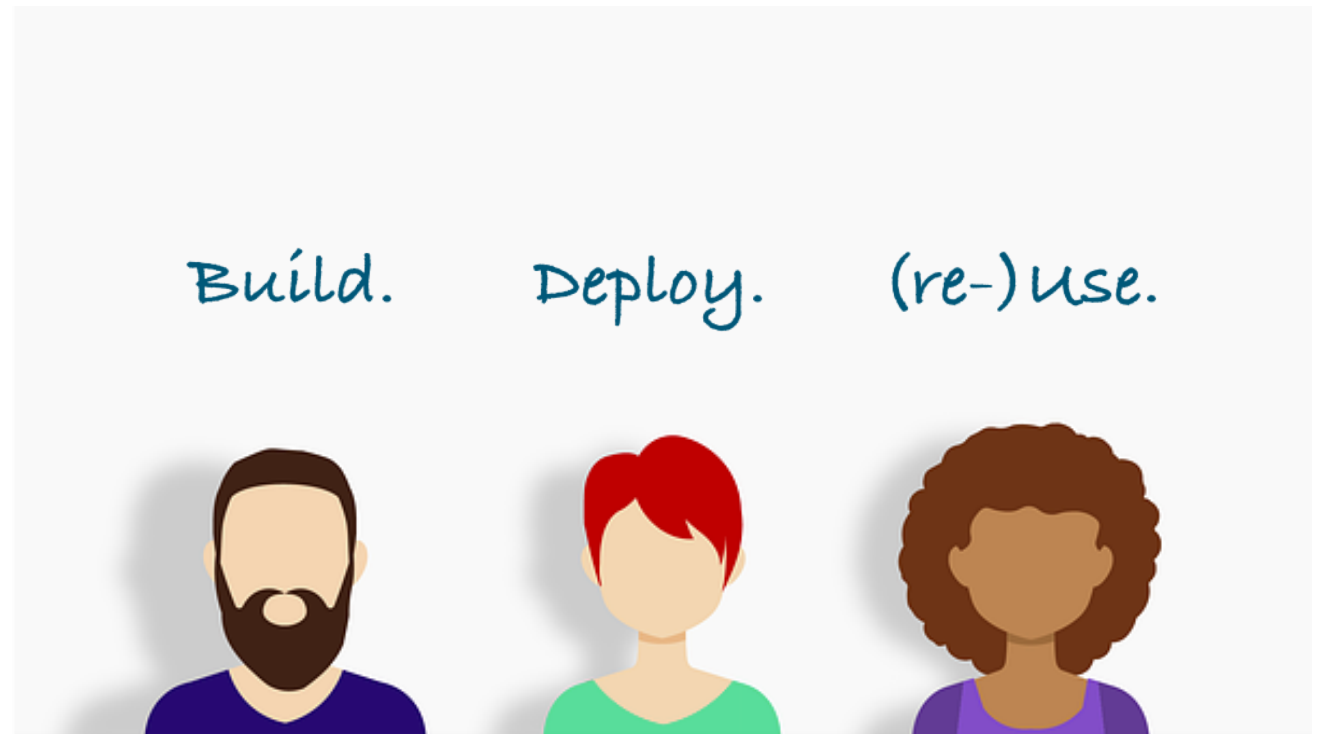
Custom component service development basics

- A custom component is simply a REST API
 - Always GET and POST which "plug" into Oracle Digital Assistant
 - Custom component typically calls an external API
- You can "package" related custom components together
 - Called a component service
- Oracle provides
 - A "starter" which has most of the code for a custom component
 - SDK to simplify bot interaction

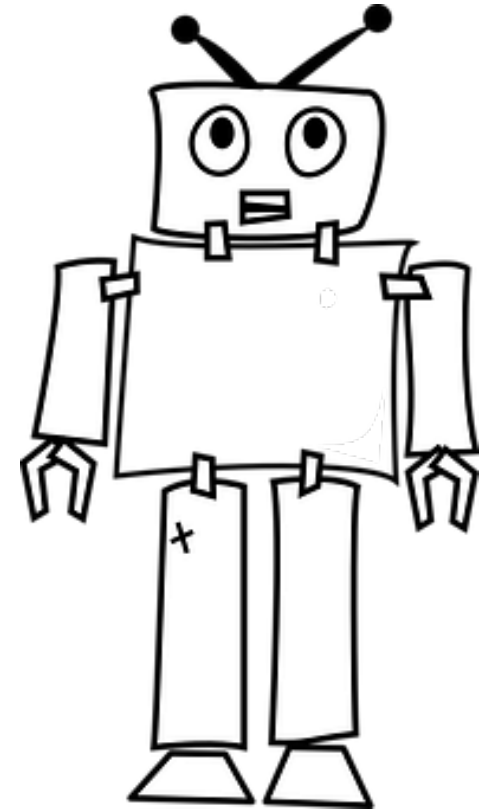


Custom component service development basics

- Most of the code is boilerplate (at least when you start)
- Exceptions are:
 - Component service name
 - Component name
 - Component properties
 - Component return action strings
 - Typically, a backend API call



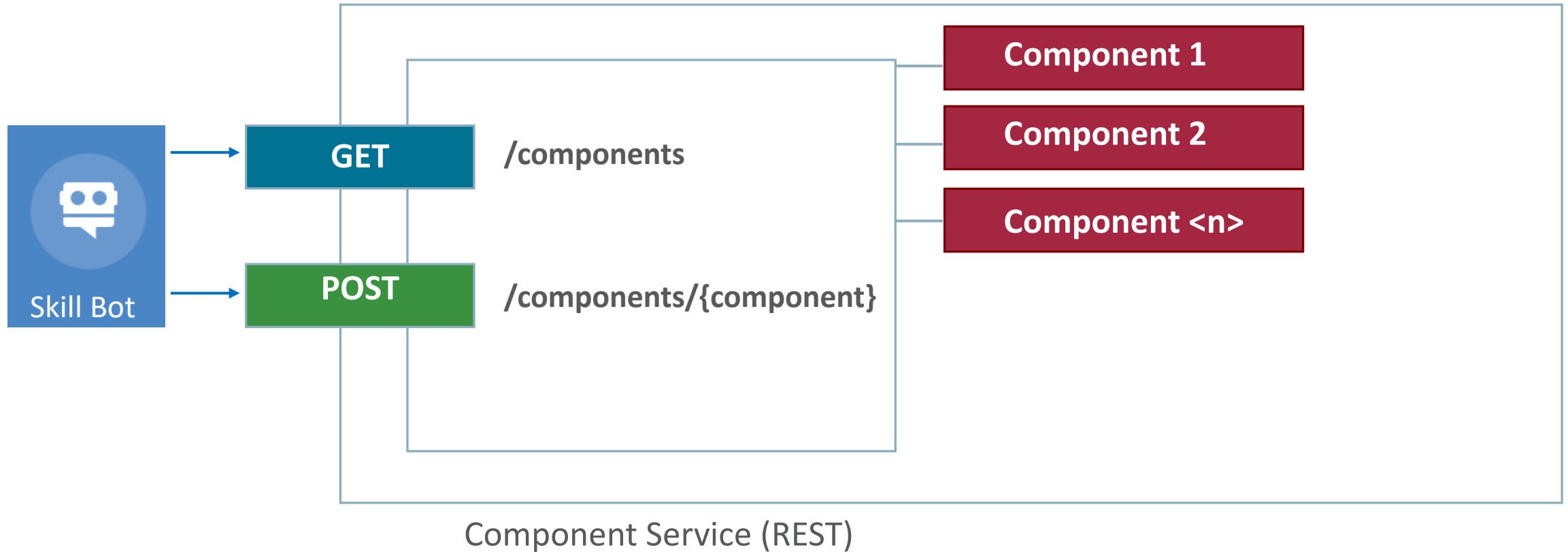
Oracle provides Oracle Bots Node.js SDK, a free utility that makes custom component development and deployment easy



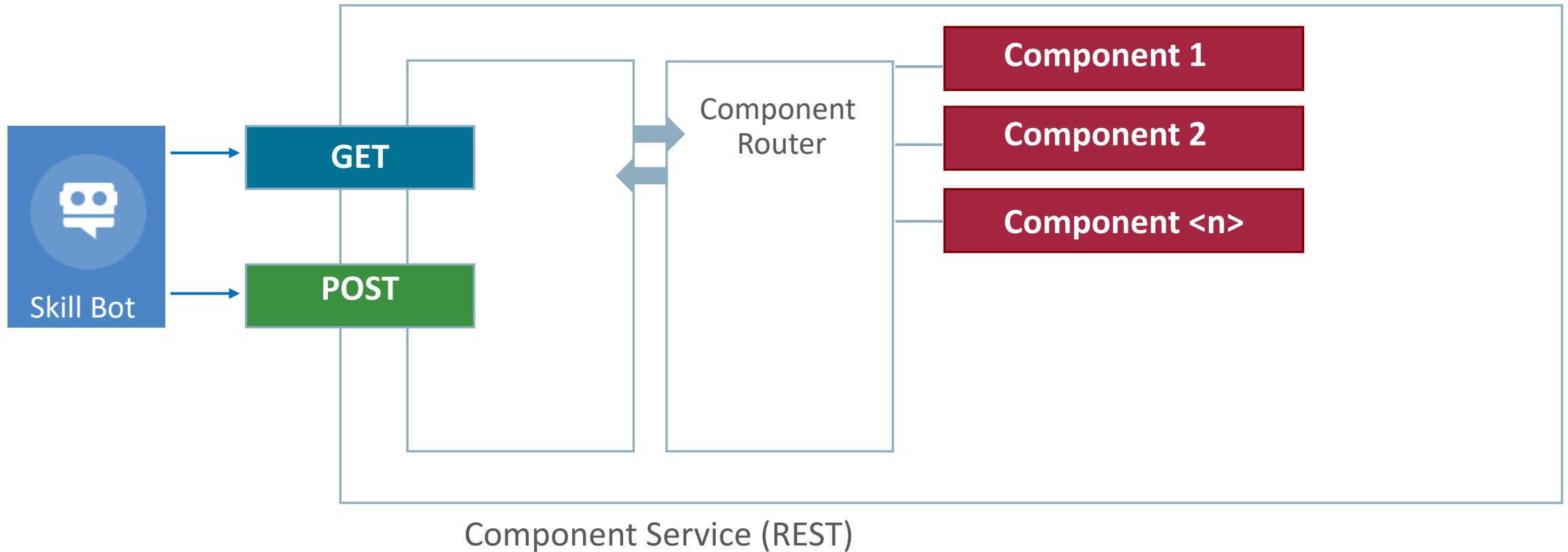
Topic agenda

- 1 Overview
- 2 Custom component architecture
- 3 Deployment options
- 4 Design considerations

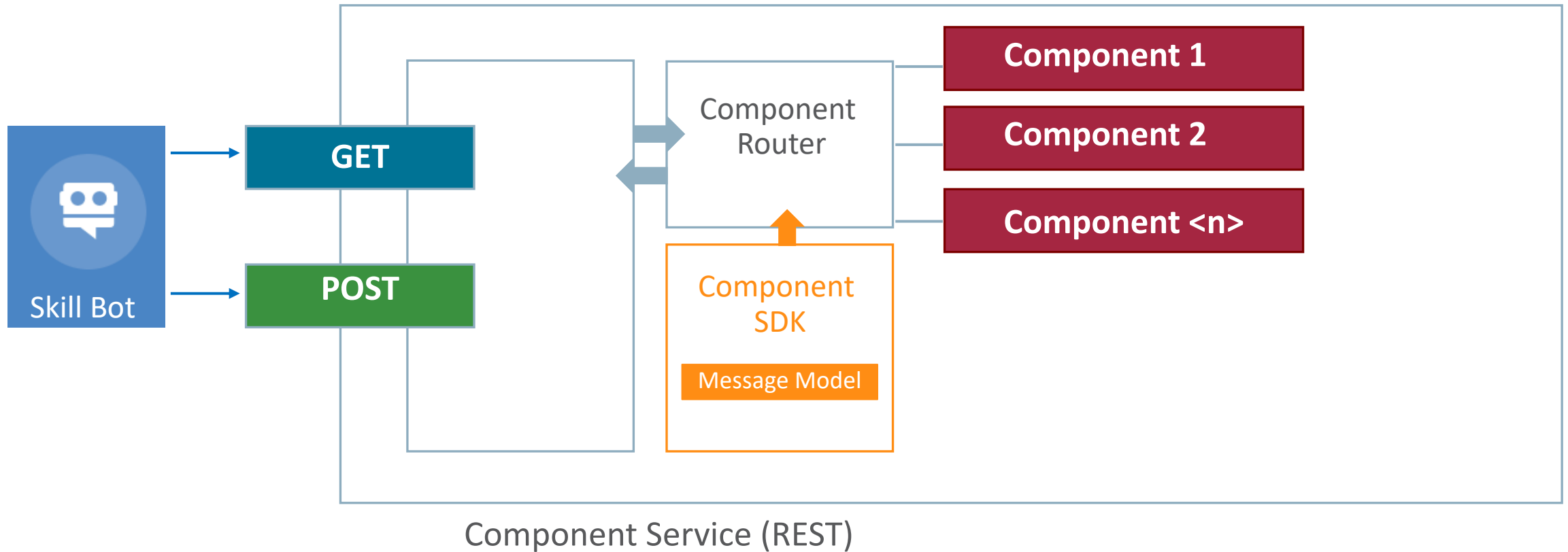
Custom component service architecture



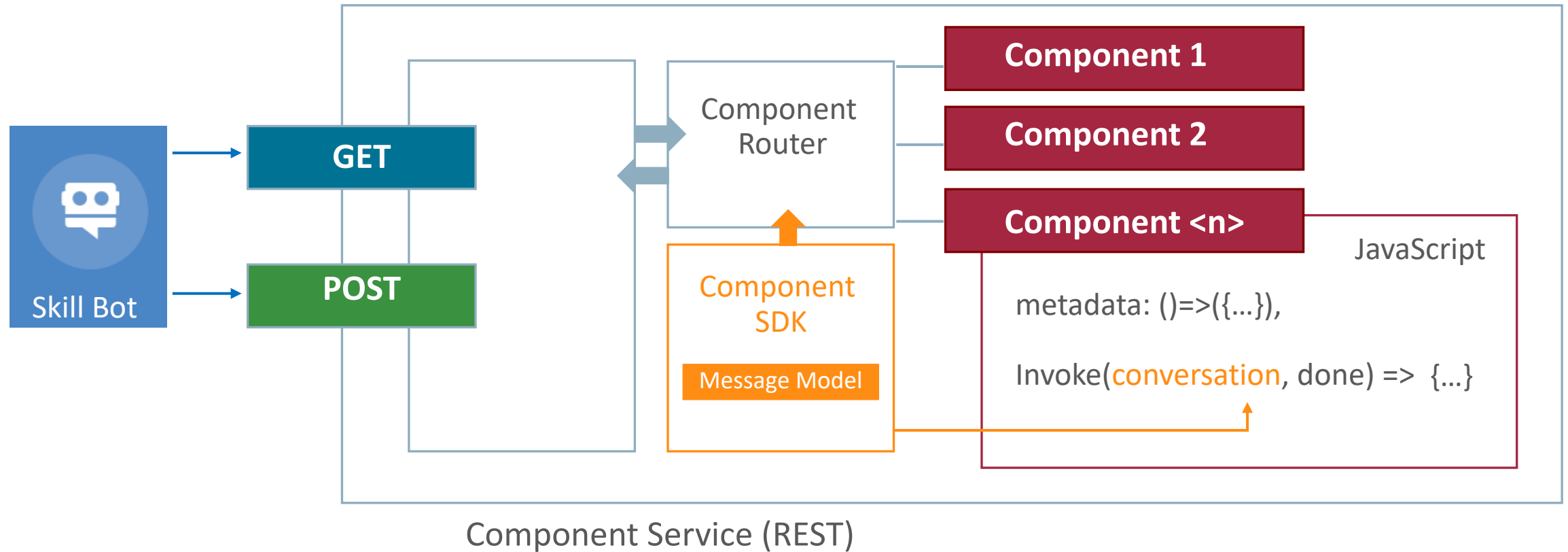
Custom component service architecture



Custom component service architecture



Custom component service architecture

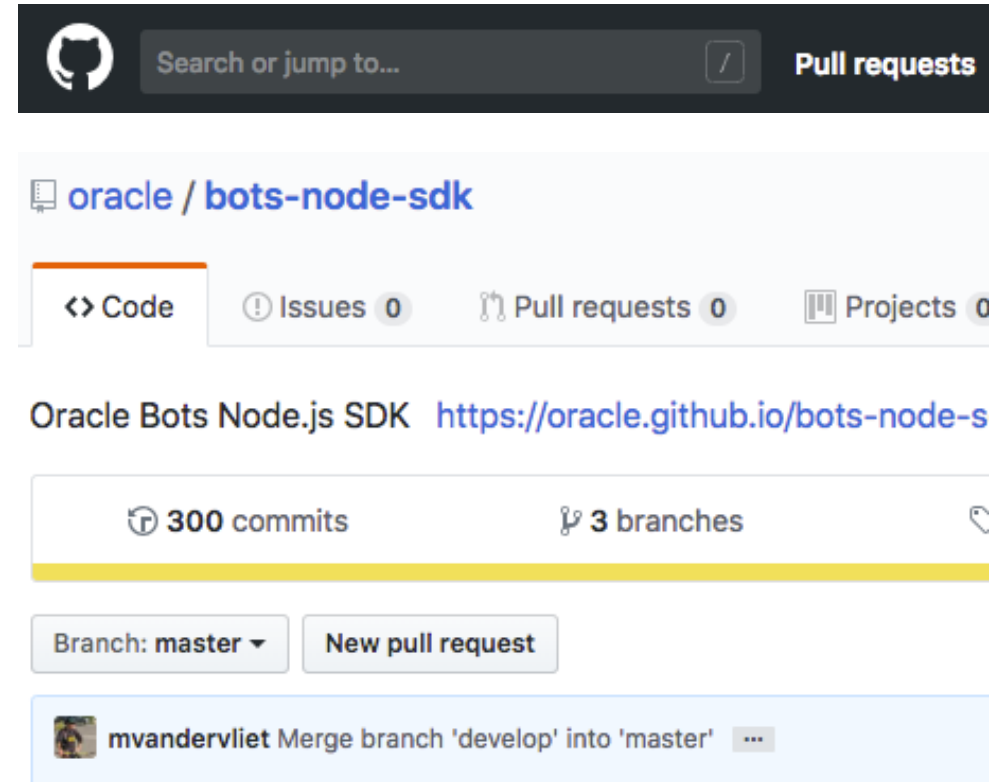


Role of the custom component SDK

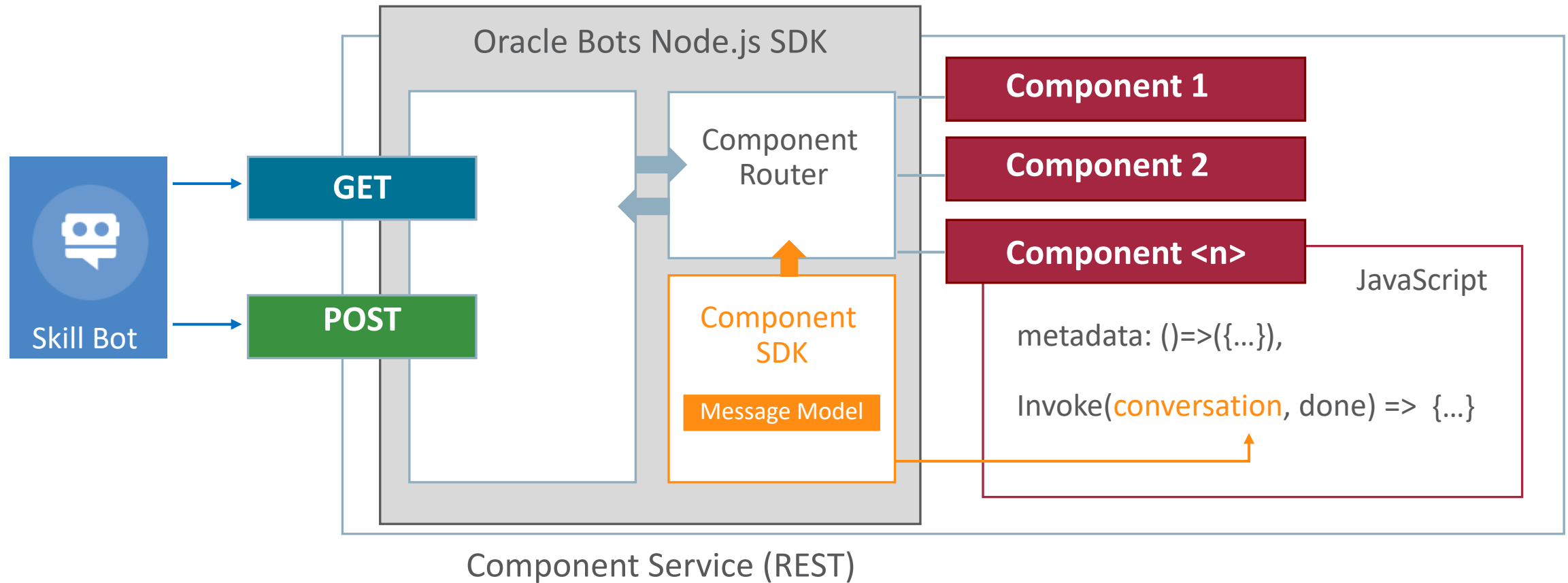
- Expose functions to
 - Read from the bot payload
 - Write to the component response payload
- Support for conversational message model responses
 - Simplifies creation of rich UI responses similar to `System.CommonResponse`
 - Supports attachments, card layouts, location, actions and more
- Supports developers detecting the type of an incoming message
 - E.g. actions need to be handled differently than text messages

Oracle Bots Node.js SDK

- Development and test tools for custom components and webhook
- Available on GitHub and NPMJS
- Requires Node Package Manager (NPM)
- Command Line Interface (CLI)
 - Creates component service and components
 - Provides custom component SDK to project
 - Integrated runtime for testing



Oracle Bots Node.js SDK



Topic agenda

- 1 Overview
- 2 Custom component architecture
- 3 Deployment options**
- 4 Design considerations

Custom component deployment options

- Local Component Container
 - Custom component service resides in skill bot
 - Single instance deployment
- Mobile Hub
 - Multi-channel environment with mobile extras
 - Mobile or multi-channel requirement
 - Shared instance deployment
- Node Container
 - If you don't need Mobile Hub but want a shared deployment

Remote deployment vs. local deployment

Remote deployment

- Pro
 - Single point of maintenance
 - Easy to push custom component changes
 - Leverage server features like access protection
- Con
 - Dependency to remote configuration needs to be tracked

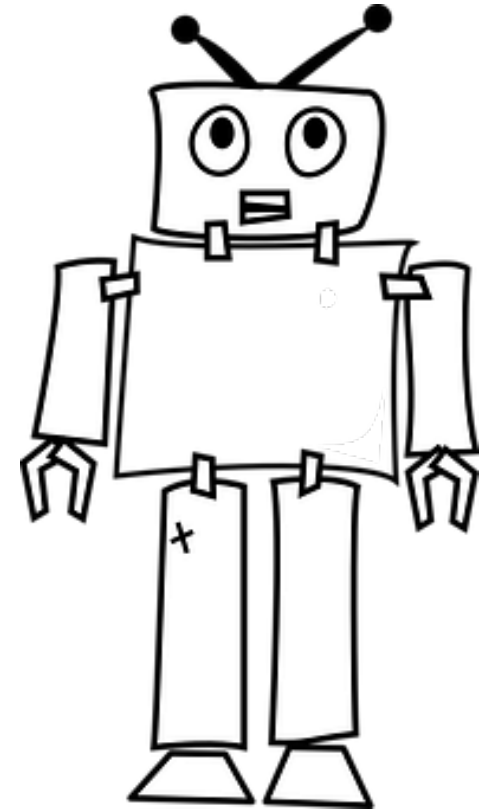
Local deployment

- Pro
 - Skill bots can run different versions of a custom component service
 - Custom component service deployed with skill bot (import / export)
 - Easier to use with skills published to skill store
- Con
 - Harder to maintain, e.g. pushing custom component changes

Topic agenda

- 1 Overview
- 2 Custom component architecture
- 3 Deployment options
- 4 Design considerations

Think of custom component services as the equivalent of **libraries** in Java



Design custom component services as libraries

- Build custom components with reuse in mind
 - Don't repeat yourself (DRY) principle
- Group related components in a single service
 - Managing user accounts
 - Transactions in a specific business area
 - Product catalog interactions
 - Utility functions
 - etc.
- A widely understood concept

Product
Component
Service

- ListProducts
- DeleteProduct
- ProductCatalog

User Action
Component
Service

- AuthenticateUser
- GetUserInfo
- SetUserInfo

Custom component naming convention

- Choose a descriptive component name
 - Good component name describes what the component does
- Use naming convention that ensure component names are unique
 - JavaScript doesn't know of name spaces
 - Expect your component to be used together with 3rd party components
 - E.g. `com.oracle.bots.financials.AccountBalance`

Integrated Cloud

Applications & Platform Services

ORACLE®



Oracle Digital Assistant Hands-On

TBD